Text Mining for Open Domain Semi-Supervised Semantic Role Labeling

Quynh Ngoc Thi Do[1], Steven Bethard[2], and Marie-Francine Moens[1]

[1] Katholieke Universiteit Leuven, Belgium
[2] University of Alabama at Birmingham, United States

Abstract. The identification and classification of some circumstance semantic roles like Location, Time, Manner and Direction, a task of Semantic Role Labeling (SRL), plays a very important role in building text understanding applications. However, the performance of the current SRL systems on those roles is often very poor, especially when the systems are applied on domains other than the ones they are trained on. We present a method to build open domain SRL system, in which the training data is expanded by replacing its predicates by words in the testing domain. A language model, which is considered as a text mining technique, and some linguistic resources are used to select from the vocabulary of the testing domain the best words for the replacement. We apply our method on the case study of transferring a semantic role labeler trained on the news domain to the children story domain. It gives us valuable improvements over the four circumstance semantic roles Location, Time, Manner and Direction.

1 Introduction

Playing an essential role in text understanding, *Semantic Role Labeling* is the task of natural language processing that specifies "Who did What to Whom, and How, When and Where?" in text [12].
For example, the processing of the sentence "Mary gave Peter a book at school yesterday" should result in the identification of a "giving" event with "Mary" as the *Agent* of the event, "Peter" as the *Recipient* and "a book" as the *Item being given*. The *Location* of the "giving" event, or where it took place, is "at school" and the *Time* of the event is "yesterday".
In this paper, we call an event ("giving" event) in a sentence the *semantic frame*, the verb or noun that evokes the frame ("gave") the *predicate*, the words ("Mary", "Peter", "a book", "at school", "yesterday") that play a role in the event the *arguments* and their roles ("Agent", "Recipient", "Thing being given", "Location", "Time") the *semantic roles*.
The task of semantic role labeling is to detect the event, to identify its arguments and assign the correct semantic roles to them. Thanks to the availability

of semantic annotated resources (e.g. PropBank[3], FrameNet[4]), supervised machine learning approaches have been very successful in constructing automatic semantic role labellers. Assuming the predicates are already given, those systems can reach an F1[5] score of 85% when the training and testing data are in the same domain. But, when testing on other domains, the scores often drop significantly[6].

Text mining is the task of automatic discovery of new, previously unknown information from unstructured document collections. Meanwhile, a language model tries to capture the properties of a language, and predicts the next word in a word sequence. It is trained on a collection of unlabeled texts, and therefore is considered as a text mining technique. Recently, we some attempts to use such language models in a semi-supervised setting for semantic recognition [5], [7], in which, other words or a statistical class of words provided by the language model, that could be exchanged at a certain position in a sentence or phrase, enriches the feature vectors used in training, or they are used to create training examples artificially. However, there is no principled way to use such language information.

In this paper, we develop a methodology to generate additional training data for SRL by replacing selected verbal predicate words in training examples using a language model. For each selected predicate in the training examples, from the vocabulary of the domain that the SRL is applied on, a list of replacement words which we believe can occur at the same position as the selected word, are generated. We introduce and explore a variety of features for identifying how words should be replaced, including predicate vs. argument status, POS, WordNet related words, and a replacement score based on a language model. As for experiment, we present a case study of improving the performance of a SRL system trained on the news domain when applying to the children story domain. The case study is based on our ongoing European project "Machine Understanding for interactive StorytElling" (MUSE)[7]. One of the fundamental goals of MUSE is to detect actors, actions, plots in children stories, and render them as 3D worlds. SRL with its function of identifying the events in texts plays an essential role in solving our problem. Among the set of semantic roles, some circumstance semantic roles like Location, Time etc. are very important to understand the full meaning of an event, while the performance of the current SRL systems on them is often very poor, especially when testing on a domain other than the one they are trained on. Thus, in our case study, we target to improve SRL on the four PropBank circumstance roles: AM-LOC (Location), AM-TMP (Time), AM-MNR (Manner) and AM-DIR (Direction).

In the next sections, we present related work (Section 2), linguistic resources

[3] http://verbs.colorado.edu/ mpalmer/projects/ace.html
[4] https://framenet.icsi.berkeley.edu
[5] Harmonic mean of recall and precision.
[6] http://ufal.mff.cuni.cz/conll2009-st/
[7] http://www.muse-project.eu/

(Section 3), underlying assumptions, objectives and task definition (Section 4), methodology (Section 5), case study (Section 6), and conclusion (Section 7).

2 Related Work

Semi-supervised approaches to semantic role labeling recently have received the attention of the computational linguistics community. Information from language models have been used as extra features to improve the performance of SRL. [17] use deep learning techniques based on semi-supervised embeddings to improve a SRL system. [3] pursue this track further and use a deep neural network architecture to obtain good word representations in the form of word-embeddings. The word embedding defines the related words which are the result of the neural network training and are usually referred to as language models. [16] use word embeddings obtained by recurrent neural networks to recover the syntactic structure of a sentence, but the method is not applied to semantic role labeling. Along these lines, a number of language models with hidden layers have been developed based on generative probabilistic approaches and applied to semantic role labeling. [5] define a latent words language model as a graphical model where at each word position in a text the distribution of exchangeable words are generated. The authors use a hidden Markov language model with dependencies defined on two previous and two following words in the discourse, and in a subsequent paper, [4] explain approximate methods to train such a model among which is Gibbs sampling. In this model, each hidden variable or latent word generates a distribution over the entire vocabulary of the training data set. The model improves the performance of SRL on the CoNLL 2008 dataset especially when few training data are given to the learner. [7] propose a hidden Markov model that learns the distribution of a hidden variable that can take K different values, and the hidden variable is dependent on the previous hidden variable in the sentence. In contrast to [5], each hidden variable can generate a span or sequence of words instead of a single word. The span contains the sequence of words for the word under consideration and the predicate. Each latent variable represents a distribution of categories of words. The model is trained with a Baum-Welch algorithm. In both [5] and [7], the respectively most probable hidden word or category of words is used as an extra feature to describe the feature vector used in the recognition. In [7], several "hidden" features are used each being the result of a different initialization of the Baum-Welch algorithm. Although appealing, these latent words language models have disadvantages. The model of [5] yields a distribution over all vocabulary words raising the need to make a selection of possibly the most probable ones when using them in the feature representation. The model of [7] relies on a fixed number of categories (or latent topics) that form the hidden variables, but it is not clear how to choose such a number especially when word spans of different sizes are used as observed variables. In this paper, we aim at using a more flexible approach where such free parameters are replaced by the use of linguistic knowledge.
Besides the semi-supervised approaches that extend the feature set of SRL, there

are other attempts to generate new training examples automatically by using unlabeled data. [6] automatically generate training examples by considering the lexical and syntactic similarity between labeled and unlabeled sentences as a graph alignment problem. use the language model of [5] to generate new training examples by replacing the head word of temporal expression training examples in the task of temporal expression recognition.
None of the above works consider both structural similarity and language models as a source of evidence for generating training examples, nor do they evaluate different approaches to similarity depending on the roles sought. In contrast to most of the above works, we evaluate the proposed methods when porting the learned model to texts from a domain that is different from the one the semantic role labeler was trained on.

3 Linguistic resources

The ***Penn Proposition Bank*** (PropBank) [13] provides a corpus annotated with semantic roles. In this resource, a semantic frame which is evoked by a verb is represented as a role set: Each role set is linked to a specific sense of the verb. Therefore, each verb has several role sets corresponding to its possible senses. The list of role sets and their semantic roles for each verb is defined in a frame file. For example, in the sentence "Mary gave Peter a book at school yesterday", the role set *give*.01 with the meaning of "transfer" evoked by the verb "give", has three main arguments $A0$, $A1$ and $A2$ that are "Agent", "Theme", and " Recipient", respectively: "[Mary *A0*] gave (*give.01*) [Peter *A2*] [a book *A1*] [at school *AM-LOC*] [yesterday *AM-TMP*]".
In ***VerbNet*** [15], English verbs are grouped into different classes, adapting the

Table 1. Main PropBank semantic roles

Role	Description
A0	Agent - extern argument
A1	Patient/Theme - intern argument
A2	Indirect object / beneficiary / instrument / attribute / end state
AM-LOC	Location (where?)
AM-TMP	Temporal marker (when?)
AM-MNR	Manner
AM-DIR	Direction

previous verbal classification of [8]. Each verbal class takes different thematic roles and certain syntactic constraints that describe their superficial behavior. VerbNet's hierarchical verb classes establish a set of possible thematic roles [8]. However, the semantic roles in VerbNet are more thematic than the ones in PropBank. For example, in VerbNet, *Agent* label is used instead of $A0$ label as in PropBank. *Patient* and *Theme* can be referred to the label $A1$ of PropBank.

In Table 2, there is an example of the information that VerbNet contains for the class $give - 13.1.1$. Members of this class share the same syntactic patterns (NP V NP PP) with corresponding thematic roles (Agent V Recipient Theme Asset). Thus, two verbs "give" and "sell" in the two sentences "Mary gave Peter a book for 20 EUR" and "Mary sold Peter a book for 20 EUR" which have the same syntactic pattern, evoke two semantic frames with the same semantic role patterns as follows:
"[Mary *Agent*] gave (*give.01*) [Peter *Recipient*] [a book *Theme*] [for 20 EUR *Asset*]"
"[Mary *Agent*] sold (*sell.01*) [Peter *Recipient*] [a book *Theme*] [for 20 EUR *Asset*]"
SemLink[8] is a project whose aim is to link together different lexical resources

Table 2. VerbNet class Give-13.1.1

Class Give-13.1.1
Roles: Agent, Theme, Recipient, Asset
Members: give, hawk, hock, lease, pawn, rent, sell
Frame: NP V NP PP.asset (*Agent* V *Recipient* *Theme* {at, for, on} *Asset*)...

via a set of mappings. These mappings will make it possible to combine the different information provided by these different lexical resources for tasks such as inferencing. The mapping between VerbNet and PropBank is available in SemLink. Each frame in PropBank is linked to a suitable VerbNet class and each role label in the PropBank frame is mapped to a VerbNet role label. Table 3 shows a mapping from the PropBank role set "give.01" to the VerbNet class "13.1.1".

WordNet [11] is a large lexical database of English. Nouns, verbs, adjectives

Table 3. Mapping from PropBank role set "give.01" to VerbNet class "13.1.1"

PropBank role set="give.01"	**VerbNet class="13.1.1"**
PropBank role label	**VerbNet role label**
A0	Agent
A1	Theme
A2	Recipient

and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. Noun and verb synsets are arranged into hierarchies. The main relation among words in WordNet is synonymy, as between the words "shut" and "close" or "car" and "automobile". Each of WordNet's 117000 synsets is linked to other synsets by means of a small number of "conceptual" relations.

[8] http://verbs.colorado.edu/semlink/

The most frequently encoded relation among synsets is the super-subordinate relation (also called hyperonymy, hyponymy or IS-A relation). It links more general synsets like *furniture*, *piece_of_furniture* to increasingly specific ones like *bed* and *bunkbed*.

4 Underlying assumptions, objectives and task definition

Semi-supervised learning is very difficult to accomplish in natural language processing tasks. In general, it is successful when the labeled training data contain seed examples that are representative for the whole data set and when the cluster hypothesis holds, that is, when a suitable similarity metric can correctly cluster the unlabeled examples with the labeled seed examples [2]. With regard to semantic role labeling, in order for the cluster hypothesis to hold, it requires that "similar" or exchangeable syntactic structures and lexical words found in the labeled and unlabeled examples cluster the linguistic phrases that form a specific semantic role.
We assume that a language model (e.g., [10], [4]) with valuable generic information on both frequent and infrequent legitimate linguistic expressions, gives us exchangeable words in context. The exchangeable words are considered as a cluster of words playing the same role on forming a specific semantic role.
In this respect, the goals of this paper are to:

- Set up a methodology for choosing unlabeled examples, guessing their labels, then using them as new training data to improve the performance of a semantic role labeler.
- Evaluate the methodology in our case study: when the SRL model is trained on news domain and applied on children story domain.

The notation of the symbols used in this paper is given in Table 4. The task

Table 4. Denotation of the symbols used in this paper.

Symbol	Meaning
$\mathbf{S}_l$	Set of manually annotated sentences
$\mathbf{S}_t$	Testing set
$\mathbf{S}_{ul}$	Set of unlabeled sentences used to train the language model
$\mathbf{S}_u$	Set of unlabeled sentences generated automatically
$\mathbf{S}_{nl}$	Set of automatically annotated semantic frames of $\mathbf{S}_u$
$\mathbf{S}_{temp}$	Set of tuples of (sentence, word to be replaced, list of replacement words)
$\mathbf{S}_{sl}$	Set of semantic frames selected for the replacement
$\mathbf{V}$	Vocabulary of $\mathbf{S}_t$
N	Maximum number of replacement words for replacement candidate
z	Context window used to calculate replacement score

of the semi-supervised semantic role labeler discussed in this paper is to learn

from a set of manually annotated sentences, a set of unannotated sentences, a language model and some linguistic resources, a model that assigns semantic roles to the set of semantic frames of sentences in a test set. A sentence may contain more than one frame. Each semantic frame consists of one predicate that evokes the frame and several arguments that play a role in the frame. Predicates and arguments may be composed of more than one word. In this paper, we work with *verbal* predicates, and use *head word labeling*, which means if an argument consists of more than one word then the semantic role is assigned to only the head word. For instance, if "in the park" is the argument playing the role AM-LOC, then only the head word of the phrase "in the park", "in", is labeled with the label AM-LOC. Given a sentence s composed of n words $w_1, w_2, ..., w_n$, for each semantic frame f in s, each word w_i ($i \in \{1, 2, ...n\}$) has received a label $r_i \in \mathbf{R} \cup \{NULL\}$ during the manual annotation for training, or will receive a label $r_i \in \mathbf{R} \cup \{NULL\}$ during testing or evaluation, where $\mathbf{R}$ is a set of predefined semantic roles and $NULL$ means empty label. If $r_i \neq NULL$, then w_i is the head of an argument of f with r_i as the semantic role. In the approach that we describe in this paper, $\mathbf{R}$ is the set of PropBank semantic roles (see Table 1).
Instead of training a SRL system on the given manually annotated sentences $\mathbf{S}_l$, we generate a set of new training examples by using a language model trained on a set of unannotated sentences, then use them together with $\mathbf{S}_l$ as the training data for the SRL.

5 Methodology

The steps of our methodology to generate new training examples and train a SRL system are shown in Figure 1.
Given a manually annotated sentence set $\mathbf{S}_l$ which can be used as training data, a set of unannotated sentences $\mathbf{S}_{ul}$, a vocabulary $\mathbf{V}$ including words in the domain of the testing data $\mathbf{S}_t$, a language model L, and some linguistic resources, we create new training examples and train the SRL system as follows: First, L is trained on $\mathbf{S}_{ul}$. Then, L, $\mathbf{S}_l$, $\mathbf{V}$ and the linguistic resources are used to generate new training examples $\mathbf{S}_{nl}$. The algorithm to generate new training examples is presented below. Finally, the SRL system is trained on $\mathbf{S}_l \cup \mathbf{S}_{nl}$.

5.1 General model for generating new training instances

In what follows, we present our general model to generate new training instances for the semantic role labeling task. The detail of our algorithm is given in Figure 1 (in dashed rectangle) and Algorithm 1. It consists of five main steps:
Step 1. Selecting data for replacement. We select a set of semantic frames from $\mathbf{S}_l$ used for the replacement. The selection can be performed in different ways depending on specific case studies. For each of the selected semantic frame, we choose its predicate as the word to be replaced. For example, given a sentence "Mary gave a book to Peter at school", the semantic frame "giving" has "gave" as predicate. "gave" is the word selected for the replacement.

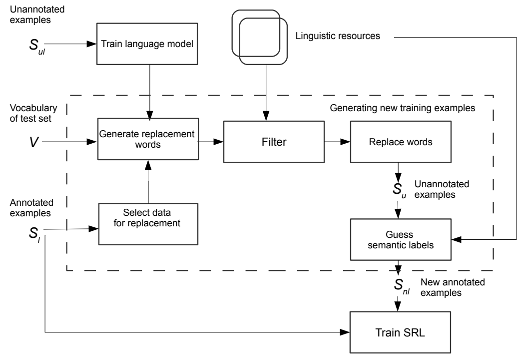

Fig. 1. Overview of the methodology to generate new training instances and train a SRL system.

Step 2. Generating replacement words for each word selected to be replaced. A statistical language model assigns a probability to a sequence of m words by means of a probability distribution. For each word selected to be replaced, we use the language model L trained on $\mathbf{S}_{ul}$, and the vocabulary $\mathbf{V}$ of the testing domain, to generate a list of replacement words. Given a sentence composed of $w_1, w_2, ..., w_n$, with w_i is the word to be replaced, for each $nw_j \in \mathbf{V}$, the score of replacing w_i by nw_j is calculated by the probability of the sequence of words $w_{i-z}, w_{i-z+1}, ..., w_{i-1}, nw_j, w_{i+1}, w_{i+2}, ..., w_{i+z}$ obtained by putting nw_j in the context of w_i where z is size of the context window taken into account:
$ReplacementScore(w_i, nw_j) = P(w_{i-z}, w_{i-z+1}, ..., w_{i-1}, nw_j, w_{i+1}, w_{i+2}, ..., w_{i+z})$
This probability score is calculated by the language model. It is used to rank the replacement words in our algorithm. Since the size of $\mathbf{V}$ may be very large and the words at the end of the list may have a very low score which often represents noise, only N words that have highest scores are chosen. After this step, we receive a ranked list of the top N replacement words for each candidate of the replacement.
Step 3. Applying filters to reduce noise in the list of replacement words. There may be a great deal of noise in the replacement words returned by the language model since it does not take into account enough information (syntactic, semantic etc.) to generate a replacement word that can be replaced perfectly for a word in a given sentence assuring the same semantic role. Thus, some linguistics filters are needed to improve the correctness and meaningfulness of the replacement.
Step 4. Replacing words in each sentence selected to be replaced by their replacement words that passed the filters, then we form a new unannotated set of sentences $\mathbf{S}_u$.
Step 5. Guessing semantic frames and their semantic labels for each sentence in $\mathbf{S}_u$ to have an annotated semantic frame set $\mathbf{S}_{nl}$.
In the following sections, we will present in more detail the language model used, some proposed filters, how to perform replacement and guess semantic role labels for the new sentences obtained by the replacement.

Algorithm 1 Generate novel training examples.

procedure GENERATENEWEXAMPLE(L, $\mathbf{S}_l$, $\mathbf{V}$, z, N)
  $\mathbf{S}_u = \emptyset$, $\mathbf{S}_{nl} = \emptyset$, $\mathbf{S}_{temp} = \emptyset$;
  Select semantic frames that are used for the replacement from $\mathbf{S}_l$: $\mathbf{S}_{sl}$ = selected semantic frames
  for each sentence $s \in \mathbf{S}_l$ **do**
    for each word w_i in s **do**
      if w_i is the predicate of a semantic frame $\in \mathbf{S}_{sl}$ **then**
        for each $nw_j \in \mathbf{V}$ **do**
          $ReplacementScore(w_i, nw_j) = P(w_{i-z}, w_{i-z+1}, ..., w_{i-1}, nw_j, w_{i+1}, w_{i+2}, ..., w_{i+z})$ obtained by using L;
        end for
        Sort nw_j according to *ReplacementScore*, then choose top N words that have highest scores forming the ranked list $\mathbf{List}_i$;
        $\mathbf{S}_{temp} = \mathbf{S}_{temp} \cup (s, w_i, \mathbf{List}_i)$;
      end if
    end for
  end for
  for each $(s, w_i, \mathbf{List}_i)$ in $\mathbf{S}_{temp}$ **do**
    for each replacement word nw_j in $\mathbf{List}_i$ **do**
      if nw_j passes **filters then**
        for each semantic frame f of s that is in $\mathbf{S}_{sl}$ and receives w_i as the predicate **do**
          s' = the sentence obtained by replacing w_i by nw_j in s;
          $\mathbf{S}_u = \mathbf{S}_u \cup s'$;
          f'= the semantic frame evoked by nw_j in s';
          Guess semantic role labels of f';
          $\mathbf{S}_{nl} = \mathbf{S}_{nl} \cup f'$;
        end for
      end if
    end for
  end for
  Return $\mathbf{S}_{nl}$
end procedure

5.2 Language model

In this paper, we use the Recurrent Neural Network Language Model[9] (RNNLM) [10] [9] which is one of the most successful techniques for statistical language modeling. Unlike previous approaches in using artificial neural networks for modeling sequential data, recurrent neural networks are not trained with limited context size. By using recurrent connections, information (e.g., words from previous sentences in a discourse) can cycle inside these networks for a long time and have an influence on the final language model obtained. The architecture of RNNLM is shown in Figure 2. The input

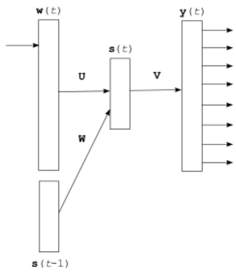

Fig. 2. Simple recurrent neural network.

layer consists of a vector $\mathbf{w}(t)$ that represents the current word w_t encoded as 1 of V with V is the vocabulary (thus size of $\mathbf{w}(t)$ is equal to the size of the vocabulary), and of vector $\mathbf{s}(t-1)$ that represents output values in the hidden layer from the previous time step. After the network is trained, the output layer $\mathbf{y}(t)$ represents $P(w_{t+1}|w_t, \mathbf{s}(t-1))$. The network is trained by stochastic gradient descent using either usual backpropagation algorithm, or backpropagation through time [14]. The network is represented by input, hidden and output layers and corresponding weight matrices - matrices $\mathbf{U}$ and $\mathbf{W}$ between the input and the hidden layer, and matrix $\mathbf{V}$ between the hidden and the output layer. Output values in the layers are computed as follows:

$$s_j(t) = f(\sum_i w_i(t)u_{ji} + \sum_l s_l(t-1)w_{jl}) \quad (1)$$

$$y_k(t) = g(\sum_j s_j(t)v_{kj}) \quad (2)$$

where *f (z)* and *g(z)* are sigmoid and softmax activation functions:

$$f(z) = \frac{1}{1+e^{-z}}, g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}} \quad (3)$$

The output layer y represents a probability distribution of the next word w_{t+1} given the history. The size of the hidden units is in our experiments set to 300. The standard backpropagation algorithm with stochastic gradient descent is used to train the model. In this research, we use the language model to calculate the probability of a word

9 http://www.fit.vutbr.cz/ imikolov/rnnlm/

sequence $W = w_1 w_2 ... w_m = W_1^m$. The language model probability of W is computed as follows:

$$P(W) = P(W_1^m) = \prod_{i=1}^{m} P(w_i | W_1^{i-1}) \tag{4}$$

Over the last few decades, an n-gram language model which assumes that the predicted word only depends on the previous n-1 words, is the most popular technique since it is simple and effective. Instead of using Equation 4, $P(W_1^m)$ is calculated in a more simple way, as $P(W_1^m) = \prod_{i=1}^{m} P(w_i | W_{i-n+1}^{i-1})$. However, an n-gram language model estimates its parameters in the discrete space, resulting in weak generalization capability on unknown data. In addition, the standard n-gram language model suffers from exponential growth of size, serious data fragmentation, and increased miss rate using longer context [18]. To overcome this problem, RNNLM, which has activation feedback with short-term memory and uses full history information instead of limiting context, can help us to calculate more accurately and efficiently $P(W_1^m)$ by Equation 4 using the output layer $y(t)$.

A language model tries to capture the properties of a language and is trained on a collection of unlabeled texts, so it can be considered as a text mining technique.

5.3 Filters

Because the list of top N replaceable words returned by the language model may contain a great deal of noise, we propose specific filters to improve the performance of the system.

Part-Of-Speech filter (POS filter): We keep replacement word nw_j for w_i if nw_j has the same POS tag as w_i, when replacing w_i in sentence s.

WordNet filter: We keep replacement word nw_j for w_i if nw_j and w_i are synonyms or have the same hypernym in WordNet. Here, we ignore the problem of word disambiguation. We only use the first word sense when finding the synonyms and the words that have the same hypernym.

For example, "January" has "Jan" as synonym, "February", "March", etc. are the words that have the same hypernym "Gregorian_calendar_month".

Predicate filter: A suitable replacement word of a predicate should also evoke a frame with correct roles when it is placed in the sentence of the target. Our idea is to assign role labels to the new frame based on the role labels of the current frame, but it raises the problem of how to find a mapping between the role sets of the two frames, and detect the correct sense of the new frame. Based on this idea, one possibility is to select only replacement words for which the mappings between role sets are available. By using SemLink (see Section 3), we define a filter specifically for predicates: for each predicate w_p evoking a frame f, we keep replacement predicate word nw_j for w_p if f and one frame evoked by nw_j are mapped to the same VerbNet class and the mappings from those two frames to the VerbNet class are defined in SemLink.

5.4 Replacing words and guessing semantic labels

Replacement words that have passed filters are used to generate new training examples. Given a semantic frame f of a sentence s composed of n words $w_1, w_2, ..., w_n$, w_p ($p \in \{1, 2, ..., n\}$) is the predicate of f, and $w_{a1}, w_{a2}, ..., w_{am}$ ($a1, a2, ..., am \in \{1, 2, ..., n\}$)

are the heads of the m arguments of f with $r_1, r_2, ..., r_m$ as semantic role labels, respectively. After the filtering step, the list of replacement words of w_p, $List_p$, includes j words $\{nw_1, nw_2, ..., nw_j\}$. For each $nw_t \in List_p$, we replace w_p by nw_t in sentence s and obtain sentence s' composed of n words $w_1, w_2, ...w_{p-1}, nw_t, w_{p+1}, ..., w_n$. If nw_t has passed the Predicate filter - which we use as a default filter -, it can be a semantic predicate and the argument structure of the frame evoked by nw_t is similar to the argument structure of the frame evoked by w_p. Thus, we guess that nw_t also invokes a semantic frame in s' with $w_{a1}, w_{a2}, ..., w_{am}$ as arguments (the new semantic frame and f - the frame evoked by w_p - have the same argument words). In order to predict the sense and role labels of the new semantic frame, we use the mappings between PropBank semantic frames and VerbNet classes that can be found in SemLink. We first find a frame f' of nw_t so that both f' and f are mapped to a same VerbNet class c. The mappings from f and f' to c are denoted by m_1 and m_2, respectively. If the Predicate filter has been applied before, f' exists and can be found in this step. Then, we can guess that f' is the new frame evoked by nw_t in s'. As for semantic role labels of f', if in f, w_{ai} ($i \in \{1, 2, ..., m\}$) with semantic role label r_i, is a circumstance role AM-s, then its role does not change in f'. That means r_i is also the role label of w_{ai} in f'. Otherwise, if the role label of w_{ai} in f is r_i, then the role label of w_{ai} in f' is $m_2^{-1}(m_1(r_i))$. For example, the sentence "Rachel wore a hat in her room" has the frame "wear.01" (wore) with "Rachel" as A0, "hat" as A1, "in" (the head of the prepostion phrase "in her room") as AM-LOC, and the predicate "wore" has "donned" as a replacement word. By replacing "wore" by "donned" in the sentence, we have a new sentence "Rachel donned a hat in her room" and "donned" evokes a new frame. In SemLink, we can find the VerbNet class "simple_dressing-41.3.1" linked to the Propbank frame "wear.01" and one PropBank frame of the predicate "don", "don.01". The role mapping between the VerbNet class and the two frames can be found in Table5. By applying our method, we have a new frame "don.01" with "Rachel" as A0 (mapped to the "Agent" VerbNet role), "hat" as A1 (mapped to the "Theme" VerbNet role), and "in" as AM-LOC (circumstance role) (See Figure 3).

Table 5. Role mapping of "simple_dressing-41.3.1" linked to both "wear.01" and "don.01"

Role of *simple_dressing-41.3.1*	Role of *wear.01*	Role of *don.01*
Agent	Arg0	Arg0
Theme	Arg1	Arg1

6 Case study

In the EU-EP7 MUSE project[10], in which KU Leuven is involved, we instantiate a virtual world with information extracted from children stories. Our fundamental goal is to introduce a new way of exploring and understanding information by "bringing text to life" through 3D interactive storytelling. Children stories will be taken initially as input, due to the relative simplicity of such stories and the relative ease of results evaluation,

[10] http://www.muse-project.eu/

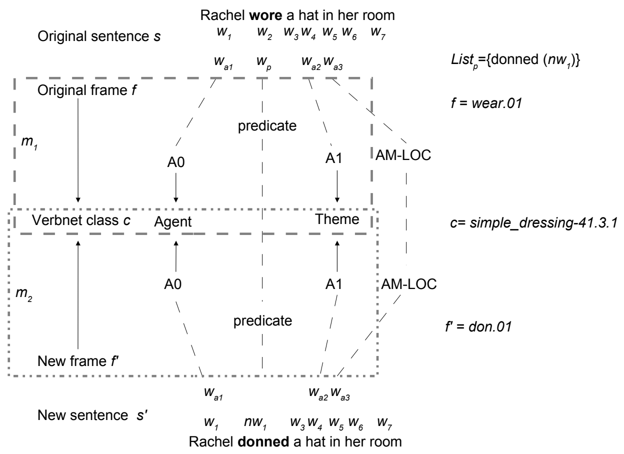

Fig. 3. An example of the replacement.

and will then be translated into formal knowledge that represents the actions, actors, plots and surrounding world. In a next step this formal knowledge is rendered as virtual 3D worlds in which the user can explore the text through interaction, re- enactment and guided game play. Finally, we will evaluate to which degree this procedure enhances the understanding by children of simple fantasy stories. In our project, NLP techniques are necessary to bring natural language closer to 3D immersive environments. Among the needed NLP techniques, SRL is one of the most important ones, since it labels the textual content of the story by semantic roles identifying actors and activities, that are then visualized in a 3D world. In this paper, we present our work on building a SRL system for the children story domain.
To create virtual 3D worlds, the location where the activities happen, the order of the activities, the tools used by the actors in the activities and the direction where the actors move toward, are very important information. Such kinds of information can be represented by the semantic roles AM-LOC, AM-TMP, AM-MNR and AM-DIR respectively in SRL. However, the performance of the current SRL systems on those roles is often very poor, especially when testing on a domain other than the one they are trained on. Therefore, in our work, we target to improve SRL system on those four roles: AM-LOC, AM-TMP, AM-MNR and AM-DIR.

6.1 Data

We select "Tuk the Hunter" story as the data for our project demonstration. The story (with some slight changes in the content) is the testing data in our case study. To evaluate the performance of SRL on our domain, we annotate semantic roles for the story following the PropBank annotation guideline and dependency head word labeling. The total number of annotated semantic frames in our testing data is 154. The detailed number of instances per role is given in Table 6.
Most of the annotated data available for semantic role labeling are in news domain. To

Table 6. Number of instances per role in the training and testing data

Data	AM-LOC	AM-TMP	AM-MNR	AM-DIR
Testing	19	28	32	10
Training	10387	23347	11837	1146

be used as our training data, we select CoNLL 2009[11] training dataset which contains parts of the Wall Street Journal corpus[12]. The detail number of instances per role in the training data is given in Table 6.

6.2 Expanding training data to children story domain

In our case study, we collect 252 children stories (mostly are fairy tales) to create the domain of children story. They are used together with the first 80 million words of the Reuters corpus to train the Recurrent Neutral Network Language Model[13], and the vocabulary of those stories is used to generate new training examples.
We realize that most of the instances of the four targeted roles are prepositions. It suggests us to choose semantic frames that contain at least one preposition argument in the CoNLL 2009 training data as the base for our training data expansion.
The SRL system is used in our experiment is the Lund university's semantic parsing [1], which is available freely, and one of the best systems in the CoNLL 2009 competition. In order to evaluate the effectiveness of our method, we compare the results obtained on the Tuk story by the Lund university's semantic parsing when training on our expanded training data and on the original training data. In our experiment setting, we use all of the three filters: POS filter, WordNet filter, Predicate filter. The maximum number of replacement words for each replacement position, $N = 500$, and the context window size, $z = 5$. Table 7 presents the results and the gains obtained on the four circumstance roles when using our expanded training data in precision, recall and F1 measures. From the table, it is clear that we obtain valuable recall, precision and F1 improvements (at least 7% for recall and F1) over all of the tested roles.

Table 7. Recall, precision, F1 results and gains (in %) per role when training on our expanded training data.

Role	Recall (Recall gain)	Precision (Precision gain)	F1 (F1 gain)
AM-LOC	47.37(+10.53)	60.00(+6.15)	52.94(+9.19)
AM-TMP	82.14(+7.14)	69.70(+7.93)	75.41(+7.67)
AM-MNR	56.25(+18.75)	85.71(+10.71)	67.92(+17.92)
AM-DIR	60.00(+10.00)	75.00(+3.57)	66.67(+7.84)

11 http://ufal.mff.cuni.cz/conll2009-st/
12 http://catalog.ldc.upenn.edu/LDC2012T04
13 http://www.fit.vutbr.cz/ imikolov/rnnlm/

7 Conclusion

In this paper, we present a methodology of building an open-domain semantic role labeling. In our case study, we transfer the SRL model trained on the news domain to the children story domain, by collecting children stories to create the new domain, then replacing verbal predicates in the training data by the words of the new domain given a language model. We keep the precision score from dropping by using the occurrence probabilities and some linguistic filters to verify linguistic patterns obtained by the replacements. The valuable enhanced results over the four circumstance roles AM-LOC, AM-TMP, AM-MNR and AM-DIR show clearly the effectiveness of our methodology on this case study[14].

Acknowledgement

This work is funded by the EU ICT FP7 FET project "Machine Understanding for interactive StorytElling" (MUSE) `http://www.muse-project.eu/`.

References

1. Björkelund, A., Hafdell, L., Nugues, P.: Multilingual semantic role labeling. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task. pp. 43–48. CoNLL '09, Association for Computational Linguistics, Stroudsburg, PA, USA (2009)
2. Chapelle, O., Scholkopf, B., Zien, A. (eds.): Semi-Supervised Learning. MIT Press (2006)
3. Collobert, R.: Deep learning for efficient discriminative parsing. In: AISTATS (2011)
4. Deschacht, K., De Belder, J., Moens, M.F.: The latent words language model. Computer Speech and Language 26(5), 384–409 (Oct 2012), `https://lirias.kuleuven.be/handle/123456789/344914`
5. Deschacht, K., Moens, M.F.: Semi-supervised semantic role labeling using the latent words language model. In: EMNLP. pp. 21–29. ACL (2009)
6. Fürstenau, H., Lapata, M.: Semi-supervised semantic role labeling via structural alignment. Comput. Linguist. 38(1), 135–171 (Mar 2012)
7. Huang, F., Ahuja, A., Downey, D., Yang, Y., Guo, Y., Yates, A.: Learning Representations for Weakly Supervised Natural Language Processing Tasks. Computational Linguistics 40, 85–120 (2013)
8. Levin, B.: English Verb Classes and Alternations A Preliminary Investigation. University of Chicago Press, Chicago and London (1993)
9. Mikolov, T.: Statistical Language Models Based on Neural Networks. Ph.D. thesis, Ph. D. thesis, Brno University of Technology (2012)
10. Mikolov, T., Karafit, M., Burget, L., Cernock, J., Khudanpur, S.: In: Kobayashi, T., Hirose, K., Nakamura, S. (eds.) INTERSPEECH. pp. 1045–1048. ISCA (2010)
11. Miller, G.A.: Wordnet: A lexical database for english. Commun. ACM 38(11), 39–41 (Nov 1995)

[14] An expanded version of this paper with comparable experimental results on the out-of-domain CoNLL 2009 testing data is submitted to the journal of IEEE/ACM Transactions on Audio, Speech and Language Processing.

12. Palmer, M., Gildea, D., Xue, N.: Semantic Role Labeling. Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers (2010)
13. Palmer, M., Kingsbury, P., Gildea, D.: The proposition bank: An annotated corpus of semantic roles. Computational Linguistics 31(1), 71–106 (2005)
14. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chap. Learning Internal Representations by Error Propagation, pp. 318–362. MIT Press, Cambridge, MA, USA (1986), http://dl.acm.org/citation.cfm?id=104279.104293
15. Schuler, K.K.: VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon. Ph.D. thesis, University of Pennsylvania (2006)
16. Socher, R., Lin, C.C.Y., Ng, A.Y., Manning, C.D.: Parsing natural scenes and natural language with recursive neural networks. In: Getoor, L., Scheffer, T. (eds.) ICML. pp. 129–136. Omnipress (2011)
17. Weston, J., Ratle, F., Collobert, R.: Deep learning via semi-supervised embedding. In: Proceedings of the 25th International Conference on Machine Learning. pp. 1168–1175. ICML '08, ACM, New York, NY, USA (2008), http://doi.acm.org/10.1145/1390156.1390303
18. Yujing Si, Zhen Zhang, T.L.J.P., Yan, Y.: Enhanced word classing for recurrent neural network language model. In: JICS: Journal of Information and Computational Science, Vol. 10. pp. 3595–3604 (2013)